IST6867 - Week 4 - Samples

John Fields

4/18/2019

Step 1: Write a summarizing function to understand the distribution of a vector

1. The function, call it 'printVecInfo' should take a vector as input
2. The function should print the following information:

a. Mean
b. Median
c. Min & max
d. Standard deviation
e. Quantiles (at 0.05 and 0.95)
f. Skewness Note for skewness, you can use the function in the 'moments' library.

3. Test the function with a vector that has (1,2,3,4,5,6,7,8,9,10,50). You should see something such as: [1] "mean: 9.54545454545454" [1] "median: 6" [1] "min: 1 max: 50" [1] "sd: 13.7212509368762" [1] "quantile (0.05 - 0.95): 1.5 – 30" [1] "skewness: 2.62039633563579"

```
#The function printVecInfo takes the information from vct4 and prints information on
#the distribution of the vector.  I tried this with print(paste...) and the cat function.
#I was able to combine in one row with the print(paste..) but couldn't format the Quantile
#values in one line as I was able to with the cat function as shown in
#https://rpubs.com/toraaglobal/sample.  NOTE: The skewness function works when running the
#printVecInfo manually but fails due to a bug when run by R Markdown.  The skewness values
#were inserted as comments after printVecInfo.
printVecInfo <- function(vct4)
{
  print(paste("Mean:",mean(vct4)))
  print(paste("Median:",median(vct4)))
  print(paste("Min:",min(vct4),"Max:",max(vct4)))
  print(paste("Standard Deviation:",sd(vct4)))
  print(paste("Quantile (.05):",quantile(vct4,probs=c(0.05))))
  print(paste("Quantile (.95):",quantile(vct4,probs=c(0.95))))
  #print(paste("Skew:",skewness(vct4)))
}
printVecInfo(vct4<-c(1:10,50))
```

```
## [1] "Mean: 9.54545454545454"
## [1] "Median: 6"
## [1] "Min: 1 Max: 50"
## [1] "Standard Deviation: 13.7212509368762"
## [1] "Quantile (.05): 1.5"
## [1] "Quantile (.95): 30"
```

```
#Skew: 2.62039633563579
```

Step 2: Creating Samples in a Jar

4. Create a variable 'jar' that has 50 red and 50 blue marbles (hint: the jar can have strings as objects, with some of the strings being 'red' and some of the strings being 'blue'

```
jar0 <- c("Red", "Blue")
jar <- rep(jar0,50)
length(jar)
```

```
## [1] 100
```

```
jar
```

```
##   [1] "Red"  "Blue" "Red"  "Blue" "Red"  "Blue" "Red"  "Blue" "Red"  "Blue"
##  [11] "Red"  "Blue" "Red"  "Blue" "Red"  "Blue" "Red"  "Blue" "Red"  "Blue"
##  [21] "Red"  "Blue" "Red"  "Blue" "Red"  "Blue" "Red"  "Blue" "Red"  "Blue"
##  [31] "Red"  "Blue" "Red"  "Blue" "Red"  "Blue" "Red"  "Blue" "Red"  "Blue"
##  [41] "Red"  "Blue" "Red"  "Blue" "Red"  "Blue" "Red"  "Blue" "Red"  "Blue"
##  [51] "Red"  "Blue" "Red"  "Blue" "Red"  "Blue" "Red"  "Blue" "Red"  "Blue"
##  [61] "Red"  "Blue" "Red"  "Blue" "Red"  "Blue" "Red"  "Blue" "Red"  "Blue"
##  [71] "Red"  "Blue" "Red"  "Blue" "Red"  "Blue" "Red"  "Blue" "Red"  "Blue"
##  [81] "Red"  "Blue" "Red"  "Blue" "Red"  "Blue" "Red"  "Blue" "Red"  "Blue"
##  [91] "Red"  "Blue" "Red"  "Blue" "Red"  "Blue" "Red"  "Blue" "Red"  "Blue"
```

5. Confirm there are 50 reds by summing the samples that are red

```
sum(jar=="Red")
```

```
## [1] 50
```

6. Sample 10 'marbles' (really strings) from the jar. How many are red? What was the percentage of red marbles?

```
#Create a new vector called firstSample for the sample of 10 'marbles' and
#then sum the "Red" and divide by the # of samples (10)
firstSample <- (sample(jar,size=10, replace=TRUE))
firstSample
```

```
##  [1] "Blue" "Blue" "Red"  "Red"  "Blue" "Blue" "Blue" "Blue" "Red"  "Red"
```

```
sum(firstSample=="Red")/10
```

```
## [1] 0.4
```

7. Do the sampling 20 times, using the 'replicate' command. This should generate a list of 20 numbers. Each number is the mean of how many reds there were in 10 samples. Use your printVecInfo to see information of the samples. Also generate a histogram of the samples.

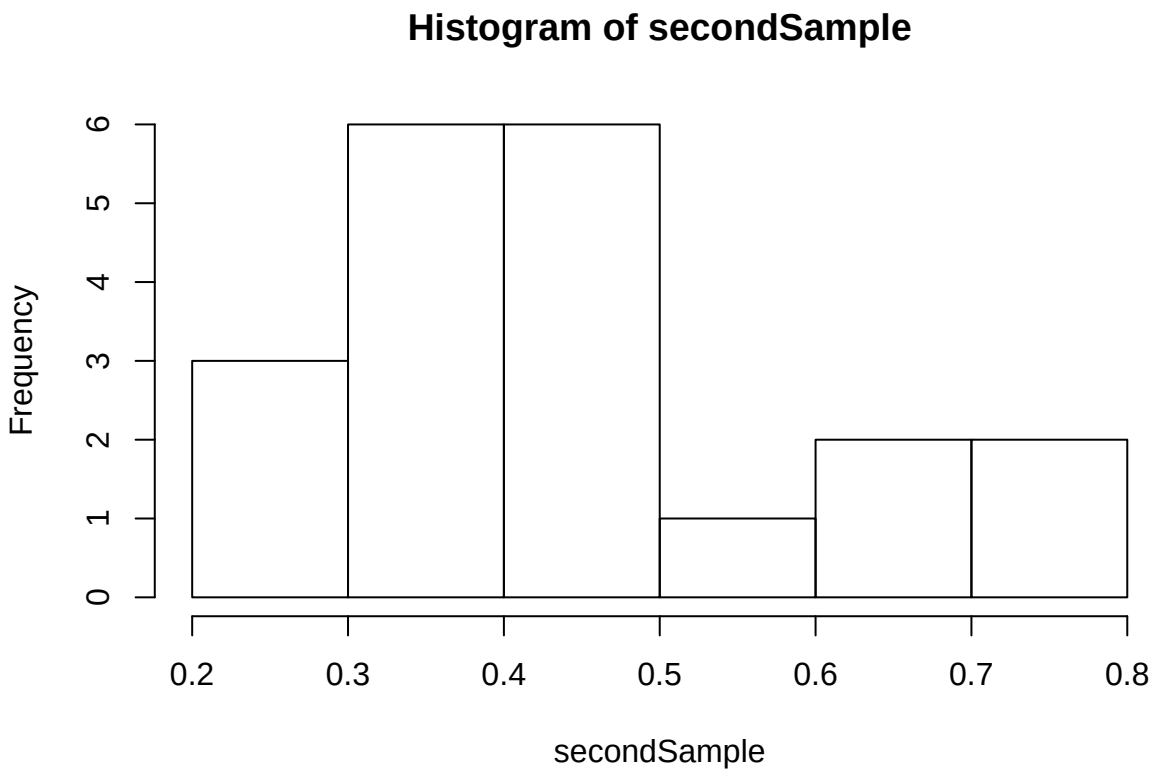
```
#Create another vector secondSample for 20 samples of 10 marbles.  At first, I struggled
#with writing the code and consulted https://rpubs.com/toraaglobal/sample for some help.

secondSample <- replicate(20,length(which((sample(jar,size=10,replace=TRUE))=="Red"))/10)
printVecInfo2 <- function(secondSample)
{
  print(paste("Mean:",mean(secondSample)))
  print(paste("Median:",median(secondSample)))
  print(paste("Min:",min(secondSample),"Max:",max(secondSample)))
  print(paste("Standard Deviation:",sd(secondSample)))
  print(paste("Quantile (.05):",quantile(secondSample,probs=c(0.05))))
  print(paste("Quantile (.95):",quantile(secondSample,probs=c(0.95))))
  #print(paste("Skew:",skewness(secondSample)))
}
secondSample
```

```
##  [1] 0.5 0.7 0.3 0.4 0.5 0.3 0.4 0.4 0.6 0.5 0.2 0.5 0.4 0.4 0.7 0.4 0.5
## [18] 0.8 0.5 0.8
```

```
printVecInfo2(secondSample)
```

```
## [1] "Mean: 0.49"
## [1] "Median: 0.5"
## [1] "Min: 0.2 Max: 0.8"
## [1] "Standard Deviation: 0.161896653195146"
## [1] "Quantile (.05): 0.295"
## [1] "Quantile (.95): 0.8"
```

```
#Skew: 0.540079418176404
hist(secondSample)
```



8. Repeat #7, but this time, sample the jar 100 times. You should get 20 numbers, this time each number represents the mean of how many reds there were in the 100 samples. Use your printVecInfo to see information of the samples. Also generate a histogram of the samples.

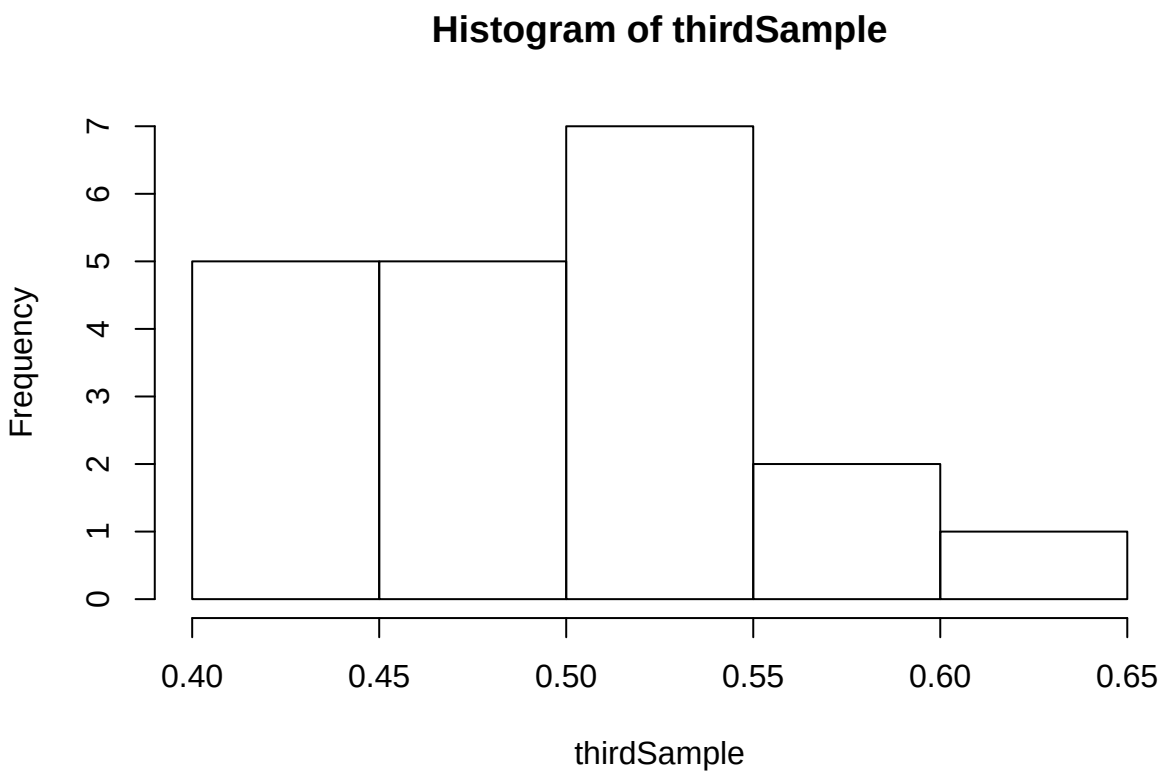
```
#Create another vector thirdSample for 100 samples of 20 marbles.
thirdSample <- replicate(20,length(which((sample(jar,size=100,replace=TRUE))=="Red"))/100)
printVecInfo3 <- function(thirdSample)
{
  print(paste("Mean:",mean(thirdSample)))
  print(paste("Median:",median(thirdSample)))
  print(paste("Min:",min(thirdSample),"Max:",max(thirdSample)))
  print(paste("Standard Deviation:",sd(thirdSample)))
  print(paste("Quantile (.05):",quantile(thirdSample,probs=c(0.05))))
  print(paste("Quantile (.95):",quantile(thirdSample,probs=c(0.95))))
  #print(paste("Skew:",skewness(thirdSample)))
}
thirdSample
```

```
##  [1] 0.47 0.52 0.52 0.57 0.55 0.49 0.53 0.40 0.49 0.48 0.54 0.40 0.44 0.46
## [15] 0.62 0.55 0.57 0.53 0.42 0.44
```

```
printVecInfo3(thirdSample)
```

```
## [1] "Mean: 0.4995"
## [1] "Median: 0.505"
## [1] "Min: 0.4 Max: 0.62"
## [1] "Standard Deviation: 0.0607388716253092"
## [1] "Quantile (.05): 0.4"
## [1] "Quantile (.95): 0.5725"
```

```
#Skew: 0.610824929571562
hist(thirdSample)
```



9. Repeat #8, but this time, replicate the sampling 100 times. You should get 100 numbers, this time each number represents the mean of how many reds there were in the 100 samples. Use your printVecInfo to see information of the samples. Also generate a histogram of the samples.

```
#Create another vector thirdSample for 100 samples of 10 marbles.
fourthSample <- replicate(100,length(which((sample(jar,size=100,replace=TRUE))=="Red"))/100)
printVecInfo3 <- function(fourthSample)
{
  print(paste("Mean:",mean(fourthSample)))
  print(paste("Median:",median(fourthSample)))
  print(paste("Min:",min(fourthSample),"Max:",max(fourthSample)))
  print(paste("Standard Deviation:",sd(fourthSample)))
  print(paste("Quantile (.05):",quantile(fourthSample,probs=c(0.05))))
  print(paste("Quantile (.95):",quantile(fourthSample,probs=c(0.95))))
  #print(paste("Skew:",skewness(fourthSample)))
```

```
}
fourthSample
```

```
##   [1] 0.50 0.50 0.52 0.42 0.53 0.48 0.53 0.43 0.51 0.50 0.49 0.52 0.47 0.52
##  [15] 0.48 0.43 0.42 0.51 0.50 0.46 0.50 0.39 0.52 0.44 0.54 0.50 0.56 0.53
##  [29] 0.52 0.56 0.54 0.53 0.47 0.52 0.52 0.45 0.55 0.46 0.49 0.49 0.56 0.48
##  [43] 0.53 0.53 0.53 0.55 0.53 0.49 0.48 0.48 0.47 0.47 0.52 0.47 0.49 0.46
##  [57] 0.46 0.59 0.54 0.44 0.40 0.49 0.45 0.46 0.51 0.52 0.44 0.46 0.42 0.50
##  [71] 0.48 0.51 0.54 0.48 0.50 0.46 0.45 0.47 0.46 0.52 0.52 0.51 0.51 0.50
##  [85] 0.56 0.49 0.43 0.40 0.46 0.44 0.52 0.46 0.51 0.47 0.46 0.49 0.50 0.51
##  [99] 0.42 0.50
```

```
printVecInfo3(fourthSample)
```

```
## [1] "Mean: 0.4905"
## [1] "Median: 0.495"
## [1] "Min: 0.39 Max: 0.59"
## [1] "Standard Deviation: 0.0401355028105864"
## [1] "Quantile (.05): 0.42"
## [1] "Quantile (.95): 0.5505"
```

```
#Skew: -0.168916618632594
hist(fourthSample)
```